Data Output Method, System and Apparatus

Field of the Invention

The present invention relates to a method, system and apparatus for outputting data to a removable storage medium and, in particular, but not exclusively to data output by printing.

Background of the Invention

A number of different techniques have been developed to minimise unauthorised access to data held on a computer apparatus or to data transmitted between computer apparatuses.

However, should a user print confidential information to a remote printer this can result in the confidential information being accessible to anyone who has access to the printer, which for mobile users can be particularly undesirable.

One solution to this problem has been to use a printer spooler, within a printer server, which will only deliver a job to a printer, for printing, if the recipients of the job authenticate themselves to the printer spooler. However, this requires specific configuration of a printer spooler, which as a result can limit the conditions under which a document can be printed.

It is desirable to improve this situation.

Embodiments of the present invention to be described hereinafter make use of a cryptographic technology known as identifier-based encryption. Accordingly, a brief description will now be given of this type of encryption.

Identifier-Based Encryption (IBE) is an emerging cryptographic schema. In this schema (see Figure 1 of the accompanying drawings), a data provider 10 encrypts payload data 13 using both an encryption key string 14, and public data 15 provided by a trusted authority12. This public data 15 is derived by

the trusted authority 12 using private data 17 and a one-way function 18. The data provider 10 then provides the encrypted payload data <13> to a recipient 11 who decrypts it, or has it decrypted, using a decryption key computed by the trusted authority 12 in dependence on the encryption key string and its own private data.

A feature of identifier-based encryption is that because the decryption key is generated from the encryption key string, its generation can be postponed until needed for decryption.

Another feature of identifier-based encryption is that the encryption key string is cryptographically unconstrained and can be any kind of string, that is, any ordered series of bits whether derived from a character string, a serialized image bit map, a digitized sound signal, or any other data source. The string may be made up of more than one component and may be formed by data already subject to upstream processing. In order to avoid cryptographic attacks based on judicious selection of a key string to reveal information about the encryption process, as part of the encryption process the encryption key string is passed through a one-way function (typically some sort of hash function) thereby making it impossible to choose a cryptographically-prejudicial encryption key string. In applications where defence against such attacks is not important, it would be possible to omit this processing of the string.

Frequently, the encryption key string serves to "identify" the intended message recipient and the trusted authority is arranged to provide the decryption key only to this identified intended recipient. This has given rise to the use of the label "identifier-based" or "identity-based" generally for cryptographic methods of the type under discussion. However, depending on the application to which such a cryptographic method is put, the string may serve a different purpose to that of identifying the intended recipient. Accordingly, the use of the term "identifier-based" or "IBE" herein in relation

to cryptographic methods and systems is to be understood simply as implying that the methods and systems are based on the use of a cryptographically unconstrained string whether or not the string serves to identify the intended recipient. Generally, in the present specification, the term "encryption key string" or "EKS" is used rather than "identity string" or "identifier string" ; the term "encryption key string" is also used in the shortened form "encryption key" for reasons of brevity.

A number of IBE algorithms are known and Figure 2 indicates, for three such algorithms, the following features, namely:

- the form of the encryption parameters 5 used, that is, the encryption key string and the public data of the trusted authority (TA);
- the conversion process 6 applied to the encryption key string to prevent attacks based on judicious selection of this string;
- the primary encryption computation 7 effected;
- the form of the encrypted output 8.

The three prior art IBE algorithms to which Figure 2 relates are:

Quadratic Residuosity (QR) method as described in the paper: C. Cocks, "An identity based encryption scheme based on quadratic residues", Proceedings of the 8th IMA International Conference on Cryptography and Coding, LNCS 2260, pp 360-363, Springer-Verlag, 2001. A brief description of this form of IBE is given hereinafter.

- **Bilinear Mappings** $\mathbb{p}$ using, for example, a Tate pairing t or modified Weil pairing $\hat{e}$. Thus, for the modified Weil pairing:

$$\hat{e}: G_1 \times G_1 \longrightarrow G_2$$

where G_1 and G_2 denote two algebraic groups of prime order q and G_2 is a subgroup of a multiplicative group of a finite field. The Tate pairing (to which the example given in Figure 2 specifically relates) can be similarly expressed though it is possible for it to be of asymmetric form:

$$t: G_1 \times G_0 \longrightarrow G_2$$

where G_0 is a further algebraic group the elements of which are not restricted to being of order q. Generally, the elements of the groups G_0 and G_1 are points on an elliptic curve though this is not necessarily the case. A description of this form of IBE method, using modified Weil pairings is given in the paper: D. Boneh, M. Franklin – "Identity-based Encryption from the Weil Pairing" in *Advances in Cryptology - CRYPTO 2001*, LNCS 2139, pp. 213-229, Springer-Verlag, 2001.

- **RSA-Based methods** The RSA public key cryptographic method is well known and in its basic form is a two-party method in which a first party generates a public/private key pair and a second party uses the first party's public key to encrypt messages for sending to the first party, the latter then using its private key to decrypt the messages. A variant of the basic RSA method, known as "mediated RSA", requires the involvement of a security mediator in order for a message recipient to be able to decrypt an encrypted message. An IBE method based on mediated RSA is described in the paper "Identity based encryption using mediated RSA", D. Boneh, X. Ding and G. Tsudik, 3rd Workshop on Information Security Application, Jeju Island, Korea, Aug, 2002.

Summary of the Invention

In accordance with a first aspect of the present invention there is provided a system comprising:

- an output device for outputting data onto a removable storage medium;
- a first computing entity arranged to encrypt a first data set based on encryption parameters comprising public data of a trusted party and an encryption key string comprising a second data set that defines a policy for allowing the output of the first data set onto a said removable storage medium, the first computing entity being further arranged to output the encrypted first data set for the output device; and
- a second computing entity associated with the trusted party and arranged when satisfied that said policy has been met, to output for the output

device a decryption key for use in decrypting the encrypted first data set, the second computing entity being arranged to generate this decryption key in dependence on the encryption key string and private data related to said public data;

the output device being arranged to use the decryption key in decrypting the encrypted first data set.

The output device is, for example, a printer.

In accordance with a second aspect of the present invention there is provided a data output method comprising the steps of:

(a) encrypting a first data set based on encryption parameters comprising public data of a trusted party and an encryption key string comprising a second data set that defines a policy for allowing the output of the first data set to a removable storage medium,

(b) providing the encrypted first data set to an output deviceadapted to output data to a removable storage medium;

(c) at the trusted party checking that said policy has been satisfied and thereafter providing the output device with a decryption key for use in decrypting the encrypted first data set, this decryption key being generated in dependence on the encryption key string and private data related to said public data; and

(d) at the output device using the decryption key in decrypting the encrypted first data set and outputting the first data set to a removable recording medium .

In accordance with a third aspect of the present invention there is provided printing apparatus including:

- means for receiving both an encryption key string comprising policy data defining a policy for allowing the printing of payload data, and said payload encrypted based on encryption parameters comprising public data of a trusted party and said encryption key string;

- means for providing the encryption key string to the trusted authority and for receiving back a decryption key; and
- means for using the received decryption key in decrypting the encrypted payload data for printing.

Brief Description of the Drawings

For a better understanding of the present invention and to understand how the same may be brought into effect reference will now be made, by way of example only, to the accompanying drawings, in which:-

. Figure 1 is a diagram illustrating the operation of a prior art encryption schema known as Identifier-Based Encryption (IBE);

. Figure 2 is a diagram illustrating how certain IBE operations are implemented by three different prior art IBE methods;

. Figure 3 is a diagram illustrating a system according to a first embodiment of the present invention;

. Figure 4 is a diagram illustrating a system according to a second embodiment of the present invention; and

. Figure 5 is a diagram illustrating a system according to a third embodiment of the present invention.

Best Mode of Carrying Out the Invention

The embodiments described below all generally provide a printing system that is arranged, using identifier based encryption, to ensure that where a job is sent to a printer, it can only be printed in cleartext if a policy associated with the job has been satisfied, this policy specifying one or more conditions, such as verification constraints to be satisfied and notifications to be made. More particularly, the job is encrypted for sending to the printer using an IBE encryption key string that is based on the policy; to decrypt the job, the printer must obtain the corresponding IBE decryption key from a trusted authority that is responsible for checking that the policy has been satisfied. As will be described below, it is possible to involve more than one trusted authority in this process, each responsible for checking that one or more conditions have

been met; in this case, the policy can be divided into sub-policies with each trusted authority only checking the relevant sub-policy, or multiple separate policies can be provided, one for each trusted authority.

First Embodiment The first embodiment is shown in Figure 3 and comprises a first computing entity 20, a second computing entity 21 and a printer 30, all connected via a network 40, for example the Internet.

The first computing entity 20 represents a user 50 and the second computing entity 21 represents a trusted authority 60.

The first and second computing entities 20, 21 are, for example, based on conventional program-controlled processors (possibly with specific hardware for implementing cryptographic processes) as are well known to a person skilled in the art. As used herein, the term "computing entity" refers to a distinct functional element but this is not to be taken as excluding the possibility of the same computer apparatus serving as the basis of two or more computing entities with the specific functionality of each such entity being provided by corresponding program processes running on the apparatus.

The first computing entity 20 includes a processor 70 that is arranged to allow the generation of a printing policy that stipulates the requirements for allowing the printing of a document, for example a policy could stipulate that a document may only be printed at a specific printer. The policy can be expressed in any suitable form, for example XML format.

Additionally or alternatively, however, the first computing entity 20 could receive the printing policy from an external source, for example, from the trusted authority 60, via the network 40.

Once the policy has been generated, or received, by the first computing entity 20 the processor 70 is arranged to use the policy, or a representation of the policy, as an encryption key string in an IBE (Identifier-Based Encryption) process for encrypting the document to be printed.

Once the document has been encrypted, it is forwarded via the network 40 to the printer 30. Typically, if the policy has been generated by the user 50, the policy is also forwarded to the printer 30 with the encrypted document.

The printer 30 includes an interface 80 for coupling the printer 30 to the network 40 and a processor 90.

Associated with the printer 30 is local printer information that includes device identity, serial number, location, etc.

On receipt of the encrypted document by the printer 30, the processor 90 is arranged, via the interface 80 and network 40, to contact the trusted authority 60 to request an associated decryption key to allow the printer 30 to decrypt the received encrypted document. Additionally, the processor 90 is arranged to forward the related printing policy to the trusted authority 60 (assuming this policy has been provided to the printer by the user 50).

On receipt by the trusted authority 60 of a request from the printer 30 for a decryption key, the trusted authority 60 determines if the trusted authority 60 has the associated policy used to derive the encryption key. The trusted authority 60 will typically receive the policy via the printer 30, as described above, however other mechanisms could be established, for example the user 50 could provide the policy to the trusted authority 60 directly. Alternatively, the trusted authority 60 could generate the relevant policy and provide it to the user 50 to allow the user 50 to encrypt the document, as described below.

On receipt of the request for a decryption key with the relevant policy, the trusted authority 60 determines whether the appropriate policy has been complied with. If the trusted authority 60 believes that the policy has been complied with, the trusted authority 60 generates an associated IBE decryption key using data corresponding to the encryption key string and forwards the decryption key to the printer 30 to enable the latter to decrypt the document. Of course, the trusted authority can generate the decryption key in parallel with, or even before, carrying out its determination as to whether the appropriate policy has been met provided that it defers providing the decryption key to the printer until satisfied that the policy has been met.

A more detailed description will now be given of the IBE encryption/ decryption processes employed by the first embodiment, these processes being based, by way of example, on the use of bilinear maps It is to be understood, however, that other IBE processes can alternatively be used such as those based on on quadratic residue techniques, or on RSA techniques.

In the following, G_1 and G_2 denote two groups of prime order q in which the discrete logarithm problem is believed to be hard and for which there exists a computable bilinear map $\hat{p}$ expressed as:

$$\hat{p} : G_1 \times G_1 \longrightarrow G_2$$

G_1 is here assumed to be a group of points on an elliptic curve (though this is not necessarily the case) and G_2 is a subgroup of a multiplicative group of a finite field $\mathbb{F}_q$. Example computable bilinear maps are the Tate pairing and the Weil pairing (though, as is well known to persons skilled in the art, for cryptographic purposes, a modified form of the Weil pairing is used that ensure $e(P,P) \neq 1$ where $P \in G_1$).

As the mapping between G_1 and G_2 is bilinear exponents/multipliers can be moved around. For example if $a, b, c \in \mathbb{F}_q$ and $P, Q \in G_1$ then

$p(aP, bQ)^c = p(aP, cQ)^b = p(bP, cQ)^a = p(bP, aQ)^c = p(cP, aQ)^b = p(cP, bQ)^a$

$= p(abP, Q)^c = p(abP, cQ) = p(P, abQ)^c = p(cP, abQ)$

$= \ldots$

$= p(abcP, Q) = p(P, abcQ) = p(P, Q)^{abc}$

To set up the system: a large (at least 512-bits) prime p is chosen such that $p = 2 \bmod 3$ and $p = 6q - 1$ for some prime $q > 3$; an elliptic curve, E, such as $y^2 = x^3 + 1$ over $\mathbb{F}_p$ is defined; and an arbitrary point, P, on E, i.e., $P \in E/\mathbb{F}_p$ of order q is chosen.

Additionally, the following cryptographic hash functions are defined:

H_1: $\{0,1\}^* \rightarrow \mathbb{F}_p$;

H_2: $\mathbb{F}_{p^2} \rightarrow \{0,1\}^k$ for some security parameter k;

H_3: $\{0,1\}^k \times \{0,1\}^k \rightarrow \mathbb{Z}^*_q$,

H_4: $\{0,1\}^k \rightarrow \{0,1\}^k$.

A public/private key pair is defined for the trusted authority 60 where the public key R is: $R \in$ G1 and the private key s is: $s \in \mathbb{F}_q$ with $R=sP \in G_1$.

Additionally, this embodiment uses an identifier based public key Q_{ID} / private key S_{ID} pair where the Q_{ID}, $S_{ID} \in G_1$ and the trusted authority's public/private key pair (R,s) is linked with the identifier based public/private key by

$S_{ID} = sQ_{ID}$ and Q_{ID} = MapToPoint (H_1 (ID))

where ID is an identifier string (encryption key string).

Given the hash function H_1: $\{0,1\}^* \rightarrow \mathbb{F}_p$, algorithm MapToPoint works as follows on input $H_1(ID) = y_0 \in \mathbb{F}_p$:

(1) Compute $x_0 = (y_0^2 - 1)^{1/3} = (y_0^2 - 1)^{(2p-1)/3} \in \mathbb{F}_p$.

(2) Let $Q = (x_0, y_0) \in E/\mathbb{F}_p$ and set $Q_{ID} = 6Q \in G_1$.

(3) Output MapToPoint(y_0) = Q_{ID}.

Identifier based encryption allows the holder of the private key S_{ID} of an identifier (encryption key string) based key pair to decrypt a document sent to them encrypted using the associated public key Q_{ID}. In the present case, the printing policy is used as the encryption key string to derive the public key Q_{ID}, hereinafter referred to as Q_{print}. Once this public key has been derived, the document m to be printed can be encrypted by performing the following computation.

- Selects a random number $\sigma \in \{0,1\}^k$.
- Computes $r = H_3(\sigma, m)$, where r is a random element that ensures only someone with the appropriate private key can decrypt the document, m.
- Computes $U = rP$.
- Computes $g_{print} = \hat{e}(Q_{print}, R) \in \mathbb{F}_{p^2}$.
- Computes $V = \sigma \oplus H_2(g_{print}^r)$.
- Computes $W = m \oplus H_4(\sigma)$.
- Sets the ciphertext to be $C = (U, V, W)$.

As stated above the ciphertext, which corresponds to the encrypted document, m, is forwarded to the printer 30.

The printer 30 contacts the trusted authority 60 to obtain the associated private key related to the public key Q_{print}. On being contacted, the trusted authority 60 checks that the printing policy on which Q_{print} is based is satisfied and, if so, provides the user 50 with the appropriate private key. The appropriate private key, here called S_{print}, is a combination of Q_{print} and the trusted authority's private key s, that is:

$$S_{print} = sQ_{print}$$

On receipt of the private key S_{print} the document is decrypted by the printer performing the following computation:

- Tests $U \in E/\mathbb{F}_p$ of order q;
- Computes $x = \hat{p}(S_{print}, U)$;
- Computes $\sigma = V \oplus H_2(x)$;
- Computes $m = W \oplus H_4(\sigma)$;
- Computes $r = H_3(\sigma, m)$;
- Checks $U = rP$.

It may be noted that in the above-noted variant where the trusted authority 60 generates the relevant policy, if the user does not need to see the policy, then the trusted authority could simply provide the user 50 with Q_{print} rather than with the underlying printing policy (encryption key string); in either case, the encryption of the documents is still based on the encryption key string and the public key of the trusted authority. Conversely, where the user 50 has generated the policy, the user can provide not only the policy but also Q_{print} to the trusted authority to save the latter having to recalculate this value; in either case, generation of the decryption key S_{print} is effected in dependence on the encryption key string and the private key of the trusted authority. In both the foregoing situations where a party (user / trusted authority) receives Q_{print} rather than the encryption key string (printing policy), that party has to trust that the link between the policy and Q_{print} has not been broken which would generally involve authentication and integrity checking with respect to the transfer of Q_{print}.

In another variant of the first embodiment the second computing entity 21 that serves as the trusted authority 60 is incorporated into a portable device 60, such as a smartcard, that can only communicate with the printer 30 when the portable device is present at the printer. More specifically, the portable device is provided with a first communications interface and the printer has a complementary second communications interface, these interfaces being

such that communication between the trusted authority and printer can only take place when the interfaces are close to each other (for example, the interfaces can be designed to require physical interconnection or to provide for a short range (<10 meters) wireless connection). In this variant the portable device would typically be carried by a person having authority to print the data of interest so that the person would need to be present at the printer before the decryption key can be provided by the trusted authority to the printer. In this case, the printing policy need not require any specific condition to be checked though, preferably, the policy at least requires that the trusted authority authenticates his/herself in some way as being the authorized possessor of the portable device (such as by input of a PIN code). In one application of this variant, the authorized possessor of the portable device can request a document to be sent by the first computing entity 20 (which may be the possessor's home computing system, for example) in encrypted form to a printer 30 near the possessor who can be anywhere in the world; in this case, only the possessor of the portable device can enable decryption of the document by the printer.

Second Embodiment The above embodiment can be expanded to include multiple trusted authorities where the decryption requires a decryption key from each of the individual trusted authorities. One embodiment of multiple trusted authorities is shown in Figure 4, which is based upon the system shown in Figure 3 with the addition of a third computing entity 100, where the third computing entity 100 acts as a second trusted authority 200, independent of the first trusted authority 60.

As with the first trusted authority 60, the second trusted authority 200 has a unique public/private key pair.

As described below, there is an independent printing policy associated with each trusted authority 60, 200, and a corresponding IBE public key Q_{print1} and Q_{print2} is formed from each policy. Each trusted authority 60, 200 generates a

private key S_{print1}, S_{print2} corresponding to the respective public key, as described above. To send an encrypted document to the printer 30 the user 50 encrypts the document with a combination of the printing-policy public keys Q_{print1}, Q_{print2} associated with the trusted authorities 60, 200 respectively, and the respective public keys R_1, R_2 of these authorities. On receipt of the encrypted document the printer 30 decrypts the document with a combination of the private keys S_{print1}, S_{print2} associated with the respective policies; the printer 30 obtains the private keys S_{print1}, S_{print2} from the trusted authorities 60, 200 respectively with each trusted authority only releasing the related private key when satisfied that the associated printing policy has been satisfied.

The second embodiment will now be described in more.

The first trusted authority 60 has a public key R_1 and a corresponding private key s_1 where $R_1 = s_1P$, with P being a point on an elliptic curve, as described above.

The second trusted authority 200 has a public key R_2 and a corresponding private key s_2 where $R_2 = s_2P$, with P being the same point on the elliptic curve as used by the first trusted authority.

The user 50 defines a first and a second printing policy that are associated with the first and second trusted authorities 60, 200 respectively, that is to say with the first trusted authority 60 the user 50 has a first policy Print1, whilst with the second trusted authority 200 the user 50 had a second policy Print2.

Using the first policy Print1 as an IBE encryption key string, a first public key Q_{print1} is derived:

$$Q_{print1} = \text{MapToPoint}H_1(\text{Print1})$$

The trusted authority 60 can use this public key to generate a corresponding IBE decryption key:

$$S_{print1} = s_1 Q_{print1}$$

Similarly, using the second policy Print2 as an IBE encryption key string, a second public key Q_{print2} is derived:

$$Q_{print2} = \text{MapToPoint}H_1(\text{Print2})$$

The trusted authority 200 can use this public key to generate a corresponding IBE decryption key:

$$S_{print2} = s_2 Q_{print2}$$

Using Q_{print1} and Q_{print2}, the user 50 encrypts a document m for sending to the printer 30 by generating ciphertext U, *V* and *W* in steps in which it:

- Selects a random number $\sigma \in \{0,1\}^k$.
- Computes $r = H_3(\sigma, m)$.
- Computes $U = rP$.
- Computes $g_{print} = \Pi_{(1 \le i \le 2)} \hat{p}(Q_{printi}, R_i) \in \mathbb{F}_{p^2}$.
- Computes $V = \sigma \oplus H_2(g_{print}^r)$.
- Computes $W = m \oplus H_4(\sigma)$.
- Sets the ciphertext to be $C = (U, V, W)$.

Decryption is performed by the printer by computing:

- Tests $U \in E/\mathbb{F}_p$ of order q;
- Computes $x = \hat{p}(\Sigma_{(1 \le i \le 2)} S_{printi}, U)$;
- Computes $\sigma = V \oplus H_2(x)$;
- Computes $m = W \oplus H_4(\sigma)$;
- Computes $r = H_3(\sigma, m)$;
- Checks $U = rP$.

where the private (decryption) keys S_{print1} and S_{print2} are provided to the printer 30 on satisfactory compliance of the respective policy Print1, Print2. As will be appreciated, the message m can only be decrypted with knowledge of both private keys S_{print1} and S_{print2}.

Figure 5 depicts a specific example of the use of two trusted authorities, one of which is associated with a computing entity provided by computer apparatus that also acts as the computing entity for the encrypting party (the user 50 of Figure 4). More particularly, Figure 5 shows a bookshop 300 that includes a printer 310; first and second computing entities 320, 321 provided on the same computing platform and respectively acting as an encrypting entity for a book publisher 330 and as a first trusted authority 340 associated with the book publisher; and a third computing entity 350 associated with the printer manufacture and also acting as a second trusted authority 360. The printer 310, the first and second computing entities 320, 321, and the second computing entity 350 are connected via a network 370, for example the Internet.

The bookshop 300 allows customers to locally print books using the printer 310. For each book, the book publisher 330 has used the computing entity 320 to provide the bookshop 300 with an encrypted version of the book encrypted using a public key derived using respective policies for the two trusted authorities 340, 360, as described above.

The first policy, intended for the first trusted authority 340 (i.e. the book publishers themselves), contains references to the book and the bookshop. The second policy requires that the second trusted authority 360 (i.e. the printer manufacture) confirm the integrity and operability of the printer 310 before issuing an appropriate private key.

When a customer attempts to print a book the printer detects the two associated policies and sends each policy to the relevant trusted authority 340, 360 to obtain the relevant private key required by the printer 310 to decrypt the book. Therefore, for a book to be printed off, the book publisher 330 can be confident that the printer integrity has been checked by the printer manufacture and that the bookshop 300 has informed the book publisher 330

that the book has been printed, thereby allowing the book publisher 300 to charge the bookshop 300 for the printed book.

It will be appreciated that the foregoing book publisher example can equally be applied to any document, not just books.

Third Embodiment This embodiment (not illustrated) further expands the printing system to involve any number n of trusted authorities. The trusted authorities can be totally independent of each other and there is no need for any business relationship to exist between the trusted authorities, in fact the trusted authorities do not need to know each other.

In this embodiment each trusted authority TA_i ($i = 1, \ldots, n$) respectively selects a random $s_i \in \mathbb{G}^*_q$ and set $R_i = s_iP$. The user encrypts a document $m \in \{0,1\}^k$ for sending to the printer 30 using n public keys $Q_{\text{Print}i}$ ($i = 1, \ldots, n$) each derived from a respective printing policy Print$i$ ($i = 1, \ldots, n$) $\in \{0,1\}^*$ that is associated with a respective one of the trusted authorities. The printer 30 can decrypt the encrypted document if the printer 30 receives the n private keys $S_{\text{Print}i}$ ($i = 1, \ldots, n$), each issued by a respective one of the trusted authorities in dependence on the associated printing policy, that is:

$$S_{\text{Print}i} = s_iQ_{\text{Print}i}.$$

More particularly, to encrypt a document, m, the user 50:

- Computes a MapToPoint $(H_1(\text{Print}i)) = Q_{\text{Print}i}$ ($i = 1, \ldots, n$) $\in E/\mathbb{F}_p$ of order q.
- Selects a random number $\sigma \in \{0,1\}^k$.
- Computes $r = H_3(\sigma, m)$, where r is a random element that ensures only someone with the appropriate private key can decrypt the document, m.
- Computes $U = rP$.

- Computes $g_{print} = \Pi_{(1 \le i \le n)} \hat{p}(Q_{printi}, R_i) \in \mathbb{F}_{p^2}$.
- Computes $V = \sigma \oplus H_2(g_{print})$.
- Computes $W = m \oplus H_4(\sigma)$.
- Sets the ciphertext to be $C = (U, V, W)$.

To decrypt the message, m, the printer 30:

- Tests $U \in E / \mathbb{F}_p$ of order q;
- Computes $x = \hat{p}(\Sigma_{(1 \le i \le n)} S_{printi}, U)$;
- Computes $\sigma = V \oplus H_2(x)$;
- Computes $m = W \oplus H_4(\sigma)$;
- Computes $r = H_3(\sigma, m)$;
- Checks $U = rP$.

It will be appreciated that many variants are possible to the above described embodiments of the invention. For example, the computing entity associated with at least one trusted authority can be incorporated into the same item of equipment as the printer itself, particularly where the role of this trusted authority is to check the integrity of the printer.

Although the above-described embodiments all concern the printing of data of interest by a printer, it will be appreciated that instead of the output of the data of interest being effected by a printer as in the all the embodiments described above, an alternative out device can be used to record the data on a removable storage medium. For example, the data of interest can be output to a device for writing to a recordable CD-ROM disc or similar optically-readable storage medium.

Where multiple trusted authorities are involved, it is possible to use a single printing policy giving rise to a single public key Q_{print} in which case computation of g_{print} in the above-described third embodiment simplifies to:

$$g_{print} = p(\Sigma_{1 \le i \le n} R_i, Q_{print})$$

Such a single printing policy is likely to be divided into a respective sub-policy (comprising one or more conditions) associated with each trusted authority, each such authority being satisfied that the policy is satisfied if its associated sub-policy is met.

Other ways of providing for the involvement of multiple trusted authorities are also possible. For example, the user can organise the document-to-be-printed as a number of data strings (say n strings) by using Shamir's secret sharing scheme, and then encrypt each string using the public data of a respective one of the trusted authorities and a corresponding printing policy. In order to recover the document in cleartext, the printer has to decrypt all of the strings by obtaining the appropriate decryption keys from the trusted authorities; it necessary to recover all strings because any n-1 strings or less cannot, according to Shamir's secret sharing scheme, disclose any information of the document. The Shamir secret sharing scheme also allows an implementation in which the participation of any *t* out of *n* share holders is sufficient to enable recovery of the secret.

In an alternative arrangement of multiple trusted authorities each associated with a respective printing policy, the user uses the data encrypted in respect of one printing policy as the data to be encrypted in respect of the next printing policy, the encrypted data resulting from the encryption effected in respect of all printing policies then being sent to the printer for decryption in successive decryption operations using decryption keys obtained from the trusted authorities.